



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1050
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/643,727	08/18/2003	James R. Kohn	1376.731US1	4686

21186 7590 02/16/2006

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH
1600 TCF TOWER
121 SOUTH EIGHT STREET
MINNEAPOLIS, MN 55402

EXAMINER

FENNEMA, ROBERT E

ART UNIT

PAPER NUMBER

2183

DATE MAILED: 02/16/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

10/643,727

Applicant(s)

KOHN, JAMES R.

Examiner

Robert E. Fennema

Art Unit

2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 18 August 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-27 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-27 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 18 August 2003 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date <u>12/9/2004</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 1-27 are pending.

Drawings

2. The drawings are objected to because they are informal. Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

3. In page 2 of the specification, in the Related Applications field, the Patent Application numbers have not been filled out. It is requested that the Applicant fills in the blanks as appropriate.

Claim Objections

4. Claim 23 is objected to for failing to end in a period. Appropriate correction is required.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

6. Claims 7, 11, 15, and 24 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. The Claims stated above all state that a sequence of values are formed by combining a portion of each respective addressing value of the first vector of addressing values to a respective one of a sequence of numbers. It is believed by the Examiner that the sequence of values is important to the functioning of the invention,

Art Unit: 2183

yet there is no disclosure of what the sequence of numbers can or has to be, or what their intended purpose is. It is therefore believed that one of ordinary skill in the art would be unable to find this sequence of numbers without large amounts of undue experimentation.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1-4, 14, 16-18, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Beard et al. (USPN 5,640,524, herein Beard), in view of Bruckert et al. (USPN 5,068,851, herein Bruckert).

9. As per Claim 1, Beard teaches: A computerized method comprising, in each of a plurality of processors including a first processor and a second processor (Column 4, Lines 42-45 disclose that while the preferred embodiment is done on a single processor, multiple processors can be used in another embodiment):

(a) loading a first vector register with addressing values (Column 2 Line 65 – Column 3 Line 1);

(b) loading a second vector register with operand values (Column 3, Lines 10-12.

The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

(d) selectively adding certain elements of the second vector of operand values based on the element addresses the duplicated values (Column 3, Lines 10-13, where addition is a common operation that can be used);

(e) loading, using indirect addressing from the first vector register, elements from memory into a third vector register (Column 3 Lines 5-10, while Column 28, Lines 18-19 show indirect addressing capabilities);

(f) adding values from the third vector register and the second vector of operand values to generate a result vector (Column 3, Lines 10-13); and

(g) storing the result vector to memory using indirect addressing (Column 3, Lines 11-13, while Column 28, Lines 18-19 show indirect addressing capabilities), but fails to teach:

(c) determining which, if any, element addresses of the first vector register have a value that duplicates a value in another element address.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed, and duplicate data would infer that something unplanned and thus incorrect has occurred. Therefore, one

of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention.

10. As per Claim 2, Beard teaches: The method of claim 1, wherein the set of operations (a),(b),(c), and (d) is performed substantially in parallel in the plurality of processors, and the set of operations (e), (f), and (g) is performed serially, one processor at a time (Column 4, Lines 39-49 teach how the "resource monitoring and dependant initiation methods" could be implemented in minimally parallel processors. It appears that steps (a) through (d) are initiation steps towards setting up a load and then a result calculation, and thus would have been capable of being performed in parallel in Beards invention. As for steps (e) through (g), they necessarily must have been performed serially in order, as each step depends on the step preceding it in order to generate a correct output).

11. As per Claim 3, Beard teaches: The method of claim 1, further comprising executing an ordered Msync operation before the set of operations (e), (f), and (g); (It is necessary for steps (e) through (g) to execute in order, in order to get correct output, and it also is necessary for steps (a) through (d) to be completed prior, in order for the correct values to be available to step (e). One of ordinary skill in the art would recognize that to do this, an Msync operation could be run before step (e) to ensure the processor is in the correct state, with the memory contained all the necessary information. See the Msync description from "The Single UNIX Specification", which

states that Msync should be used by programs that require a memory object to be in a known state) and

executing an end ordered Msync operation after the set of operations (e), (f), and (g) (After all the required steps are completed, one of ordinary skill in the art would have recognized the need to write that state to memory, so that any other processes (using the other processors in the multi-processor environment) waiting for that data can have it made available to them).

12. As per Claim 4, Beard teaches: The method of claim 3, wherein the set of operations (a), (b), (c), and (d) is performed substantially in parallel in the plurality of processors (Column 4, Lines 39-49 teach how the “resource monitoring and dependant initiation methods” could be implemented in minimally parallel processors. It appears that steps (a) through (d) are initiation steps towards setting up a load and then a result calculation, and thus would have been capable of being performed in parallel in Beards invention).

13. As per Claim 14, Beard teaches: A system comprising;
a first vector register having addressing values (Column 2 Line 65 – Column 3 Line 1);

a second vector register having operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

circuitry programmed to selectively add certain elements of the second vector of operand values based on the element addresses the duplicated values (Column 3, Lines 10-13, where addition is a common operation that can be used);

circuitry programmed to load, using indirect addressing from the first vector register, elements from memory into a third vector register (Column 3 Lines 5-10, while Column 28, Lines 18-19 show indirect addressing capabilities);

circuitry programmed to add values from the third vector register and the second vector of operand values to generate a result vector (Column 3, Lines 10-13); and

circuitry programmed to store the result vector to memory using indirect addressing, but fails to teach (Column 3 Lines 11-13, while Column 28, Lines 18-19 show indirect addressing capabilities):

circuitry programmed to determine which, if any, element addresses of the first vector register have a value that duplicates a value in another element address.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed, and duplicate data would infer that something unplanned and thus incorrect has occurred. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention.

14. As per Claim 16, Beard teaches: The system of claim 14, further comprising:

circuitry programmed to perform the set of operations (a), (b), (c), and (d) substantially in parallel in the plurality of processors, and

circuitry programmed to perform the set of operations (e), (f), and (g) serially, one processor at a time (Column 4, Lines 39-49 teach how the “resource monitoring and dependant initiation methods” could be implemented in minimally parallel processors. It appears that steps (a) through (d) are initiation steps towards setting up a load and then a result calculation, and thus would have been capable of being performed in parallel in Beards invention. As for steps (e) through (g), they necessarily must have been performed serially in order, as each step depends on the step preceding it in order to generate a correct output).

17. The system of claim 14, further comprising:

circuitry programmed to execute an ordered Msync operation before the set of operations (e), (f), and (g) (It is necessary for steps (e) through (g) to execute in order, in order to get correct output, and it also is necessary for steps (a) through (d) to be completed prior, in order for the correct values to be available to step (e). One of ordinary skill in the art would recognize that to do this, an Msync operation could be run before step (e) to ensure the processor is in the correct state, with the memory contained all the necessary information. See the Msync description from “The Single UNIX Specification”, which states that Msync should be used by programs that require a memory object to be in a known state); and

circuitry programmed to execute an end ordered Msync operation after the set of operations (e), (f), and (g) (After all the required steps are completed, one of ordinary skill in the art would have recognized the need to write that state to memory, so that any other processes (using the other processors in the multi-processor environment) waiting for that data can have it made available to them).

15. As per Claim 18, Beard teaches: The system of claim 17, further comprising:
circuitry programmed to perform the set of operations (a), (b), (c), and (d) substantially in parallel in the plurality of processors (Column 4, Lines 39-49 teach how the "resource monitoring and dependant initiation methods" could be implemented in minimally parallel processors. It appears that steps (a) through (d) are initiation steps towards setting up a load and then a result calculation, and thus would have been capable of being performed in parallel in Beards invention).

16. As per Claim 27, Beard teaches: A computer-readable medium having instructions stored thereon for causing a suitably programmed information-processing system to execute a method comprising:

loading a first vector register with addressing values (Column 2 Line 65 – Column 3 Line 1);

loading a second vector register with operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

selectively adding certain elements of the second vector of operand values based on the element addresses the duplicated values (Column 3, Lines 10-13, where addition is a common operation that can be used);

loading, using indirect addressing from the first vector register, elements from memory into a third vector register (Column 3 Lines 5-10, while Column 28, Lines 18-19 show indirect addressing capabilities);

adding values from the third vector register and the second vector of operand values to generate a result vector (Column 3, Lines 10-13); and

storing the result vector to memory using indirect addressing (Column 3 Lines 11-13, while Column 28, Lines 18-19 show indirect addressing capabilities), but fails to teach:

determining which, if any, element addresses of the first vector register have a value that duplicates a value in another element address.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed, and duplicate data would infer that something unplanned and thus incorrect has occurred. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention.

17. Claims 5-6, 8-11, 13, 19-24, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Beard and Bruckert, in view of Patterson et al. (herein Patterson).

18. As per Claim 5, Beard teaches the method of claim 1, but fails to teach: further comprising

executing a first barrier synchronization operation before the set of operations (e), (f), and (g) in all of the plurality of processors;

executing a second barrier synchronization operation before the set of operations (e), (f) and (g) in the second processor;

executing the set of operations (e), (f), and (g) in the first processor and then executing a second barrier synchronization operation in the first processor to satisfy the second barrier synchronization in the second processor, and executing a third barrier synchronization in the first processor; and

executing the set of operations (e), (f), and (g) in the second processor and then executing a third barrier synchronization operation in the second processor to satisfy the third barrier synchronization in the first processor.

It appears necessary that steps (a) through (d) be completely prior to steps (e) through (g) executing, and that steps (e) through (g) be done in order, in order to generate the correct output. Patterson teaches that in multiprocessor systems running parallel operations on the same data, a barrier is a common synchronization technique to force concurrently running processes to wait for previous instructions (or steps) to complete. One of ordinary skill in the art would have recognized this method as a

Art Unit: 2183

common way to synchronize multiple processors working on these steps, and how to use them. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use a barrier to ensure that the steps were done in the appropriate order.

19. As per Claim 6, Beard teaches: The method of claim 5, wherein the set of operations (a), (b), (c), and (d) is performed substantially in parallel in the plurality of processors (Column 4, Lines 39-49 teach how the "resource monitoring and dependant initiation methods" could be implemented in minimally parallel processors. It appears that steps (a) through (d) are initiation steps towards setting up a load and then a result calculation, and thus would have been capable of being performed in parallel in Beards invention).

20. As per Claim 8, Beard teaches: A computerized method comprising:

(a) within a first vector processor:

loading a first vector register in the first vector processor with addressing values (Column 2 Line 65 – Column 3 Line 1);

loading a second vector register in the first vector processor with operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

selectively adding certain elements of the second vector of operand values in the first vector processor based on the element addresses the duplicated values (Column 3, Lines 10-13, where addition is a common operation that can be used);

(b) within a second vector processor:

loading a first vector register in the second vector processor with addressing values (Column 2 Line 65 – Column 3 Line 1);

loading a second vector register in the second vector processor with operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

selectively operating on certain elements of the second vector of operand values in the second vector processor based on the element addresses the duplicated values (Column 3, Lines 10-13, where addition is a common operation that can be used);

(d) within the first vector processor:

loading, using indirect addressing from the first vector register, elements from memory into a third vector register in the first vector processor (Column 3 Lines 5-10, while Column 28, Lines 18-19 show indirect addressing capabilities);

operating on values from the third vector register and the second vector of operand values in the first vector processor to generate a first result vector (Column 3, Lines 10-13); and

storing the first result vector to memory using indirect addressing (Column 3, Lines 11-13, while Column 28, Lines 18-19 show indirect addressing capabilities);

(f) within the second vector processor:

loading, using indirect addressing from the first vector register, elements from memory into a third vector register in the second vector processor (Column 3 Lines 5-10, while Column 28, Lines 18-19 show indirect addressing capabilities);

operating on values from the third vector register and the second vector of operand values in the second vector processor to generate a second result vector (Column 3, Lines 10-13); and

storing the second result vector to memory using indirect addressing (Column 3, Lines 11-13, while Column 28, Lines 18-19 show indirect addressing capabilities), but fails to teach:

determining which, if any, element addresses of the first vector register in the first vector processor have a value that duplicates a value in another element address;

determining which, if any, element addresses of the first vector register in the second vector processor have a value that duplicates a value in another element address;

(c) performing a synchronization operation that ensures that prior store operations effectively complete in at least the second vector processor before the following (d) operation;

(e) performing a synchronization operation that ensures that the storing of the first result vector effectively completes before the following (f) operations.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is

Art Unit: 2183

being able to recognize if an instruction(s) need to be re-executed, and duplicate data would infer that something unplanned and thus incorrect has occurred. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention.

Patterson teaches that in multiprocessor systems running parallel operations on the same data, a barrier is a common synchronization technique to force concurrently running processes to wait for previous instructions (or steps) to complete. One of ordinary skill in the art would have recognized this method as a common way to synchronize multiple processors working on these steps, and how to use them. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use a barrier to ensure that the steps were done in the appropriate order.

21. As per Claim 9, Beard teaches: The method of claim 8, wherein each of the operating on functions includes adding (Column 11, Line 32, where an adder adds).

22. As per Claim 10, Beard teaches: The method of claim 9, wherein the adding includes a floating-point addition operation that produces at least one element of the result vector as an ordered- operation floating point summation of an element of the loaded third vector register and a plurality of respective elements of the original second vector of operand values corresponding to elements of the first vector of addressing

Art Unit: 2183

values having identical values (Column 4 Line 65 – Column 5 Line 19 shows floating point to be one of the available operations. Also see Column 3, Lines 10-13).

23. As per Claim 11, Beard teaches: The method of claim 8,

wherein the determining of duplicates includes:

generating each respective address value for a sequence of addressed locations within a constrained area of memory containing 2^N consecutive addresses using an N-bit value derived from each respective addressing value of the first vector register (Column 7 Lines 30-36),

generating each respective data value of a first sequence of values by combining at least a portion of each respective addressing value of the first vector register to a respective one of a sequence of integer numbers (It is a common practice in computing to “sign extend” values, when their size is not the appropriate size to fit into memory.

This entails prefixing the value with 1’s or 0’s as appropriate. It is likely that data needing to be stored after an operation would need to be sign extended in order to properly fit in the memory, thus requiring the concatenation of some value with a 0 or 1.

See Computer Arithmetic Lecture, Pages 11-12),

storing the first sequence of values to the constrained memory area using the generated sequence of respective address values (Column 3, Lines 19-22),

loading a second first sequence of values from the constrained memory area using the generated sequence of respective address values (Column 3, Lines 5-10), but fails to teach:

comparing the first sequence of values to the second sequence of values.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention.

24. As per Claim 13, Beard teaches: The method of claim 8, further wherein indirect addresses of the elements from memory are calculated by adding each respective addressing value to a base address (Column 3 Lines 2-5).

25. As per Claim 19, Beard teaches the system of claim 14, but fails to teach: further comprising:

circuitry programmed to execute a first barrier synchronization operation before the set of operations (e), (f), and (g) in all of the plurality of processors;

circuitry programmed to execute a second barrier synchronization operation before the set of operations (e), (f), and (g) in the second processor;

circuitry programmed to execute the set of operations (e), (f), and (g) in the first processor and then executing a second barrier synchronization operation in the first processor to satisfy the second barrier synchronization in the second processor, and executing a third barrier synchronization in the first processor; and

circuitry programmed to execute the set of operations (e), (f), and (g) in the second processor and then executing a third barrier synchronization operation in the second processor to satisfy the third barrier synchronization in the first processor.

It appears necessary that steps (a) through (d) be completely prior to steps (e) through (g) executing, and that steps (e) through (g) be done in order, in order to generate the correct output. Patterson teaches that in multiprocessor systems running parallel operations on the same data, a barrier is a common synchronization technique to force concurrently running processes to wait for previous instructions (or steps) to complete. One of ordinary skill in the art would have recognized this method as a common way to synchronize multiple processors working on these steps, and how to use them. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use a barrier to ensure that the steps were done in the appropriate order.

26. As per Claim 20, Beard teaches: The system of claim 19, further comprising circuitry programmed to perform the set of operations (a), (b), (c), and (d) substantially in parallel in the plurality of processors (Column 4, Lines 39-49 teach how the "resource monitoring and dependant initiation methods" could be implemented in minimally parallel processors. It appears that steps (a) through (d) are initiation steps towards setting up a load and then a result calculation, and thus would have been capable of being performed in parallel in Beards invention).

27. As per Claim 21, Beard teaches: A system comprising:

(a) a first vector processor that includes:

means for loading a first vector register in the first vector processor with addressing values (Column 2 Line 65 – Column 3 Line 1);

means for loading a second vector register in the first vector processor with operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

means for selectively adding certain elements of the second vector of operand values in the first vector processor based on the element addresses the duplicated values (Column 3, Lines 10-13, where addition is a common operation that can be used);

(b) a second vector processor that includes:

means for loading a first vector register in the second vector processor with addressing values (Column 2 Line 65 – Column 3 Line 1);

means for loading a second vector register in the second vector processor with operand values (Column 3, Lines 10-12. The second vector instruction would necessarily need a vector of operand values in order to operate on the retrieved data words);

means for selectively operating on certain elements of the second vector of operand values in the second vector processor based on the element addresses the

Art Unit: 2183

duplicated values (Column 3, Lines 10-13, where addition is a common operation that can be used);

(d) within the first vector processor;

means for loading, using indirect addressing from the first vector register, elements from memory into a third vector register in the first vector processor (Column 3 Lines 5-10, while Column 28, Lines 18-19 show indirect addressing capabilities);

means for operating on values from the third vector register and the second vector of operand values in the first vector processor to generate a first result vector (Column 3, Lines 10-13); and

means for storing the first result vector to memory using indirect addressing;

(f) within the second vector processor (Column 3, Lines 11-13, while Column 28, Lines 18-19 show indirect addressing capabilities):

means for loading, using indirect addressing from the first vector register, elements from memory into a third vector register in the second vector processor (Column 3 Lines 5-10, while Column 28, Lines 18-19 show indirect addressing capabilities);

means for operating on values from the third vector register and the second vector of operand values in the second vector processor to generate a second result vector (Column 3, Lines 10-13); and

means for storing the second result vector to memory using indirect addressing (Column 3, Lines 11-13, while Column 28, Lines 18-19 show indirect addressing capabilities), but fails to teach:

means for determining which, if any, element addresses of the first vector register in the first vector processor have a value that duplicates a value in another element address;

means for determining which, if any, element addresses of the first vector register in the second vector processor have a value that duplicates a value in another element address;

(c) means for performing a synchronization operation that ensures that prior store operations effectively complete in at least the second vector processors before the operations of the following (d) means;

(e) performing a synchronization operation that ensures that the storing of the first result vector effectively completes before the operations of the following (f) means. Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed, and duplicate data would infer that something unplanned and thus incorrect has occurred. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention.

Patterson teaches that in multiprocessor systems running parallel operations on the same data, a barrier is a common synchronization technique to force concurrently running processes to wait for previous instructions (or steps) to complete. One of ordinary skill in the art would have recognized this method as a common way to

synchronize multiple processors working on these steps, and how to use them.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use a barrier to ensure that the steps were done in the appropriate order.

28. As per Claim 22, Beard teaches: The system of claim 21, wherein each of the means for operating on functions includes an adder (Column 11, Line 32).

29. As per Claim 23, Beard teaches: The system of claim 22, further wherein the adder includes a floating-point adder that produces at least one element of the result vector as an ordered-operation floating point summation of an element of the loaded third vector register and a plurality of respective elements of the original second vector of operand values corresponding to elements of the first vector of addressing values having identical values (Column 4 Line 65 – Column 5 Line 19 shows floating point to be one of the available operations. Also see Column 3, Lines 10-13).

30. As per Claim 24, Beard teaches: The system of claim 23, wherein the means for determining of duplicates further includes:

means for generating each respective address value for a sequence of addressed locations within a constrained area of memory containing 2^N consecutive addresses using an N-bit value derived from each respective addressing value of the first vector register (Column 7 Lines 30-36),

means for generating each respective data value of a first sequence of values by combining at least a portion of each respective addressing value of the first vector register to a respective one of a sequence of integer numbers (It is a common practice in computing to "sign extend" values, when their size is not the appropriate size to fit into memory. This entails prefixing the value with 1's or 0's as appropriate. It is likely that data needing to be stored after an operation would need to be sign extended in order to properly fit in the memory, thus requiring the concatenation of some value with a 0 or 1. See Computer Arithmetic Lecture, Pages 11-12),

means for storing the first sequence of values to the constrained memory area using the generated sequence of respective address values (Column 3, Lines 19-22),

means for loading a second first sequence of values from the constrained memory area using the generated sequence of respective address values (Column 3, Lines 5-10), and

means for comparing the first sequence of values to the second sequence of values (Column 18, Lines 40-44 show a compare circuit).

31. As per Claim 26, Beard teaches: The system of claim 21, further wherein indirect addresses of the elements from memory are calculated by adding each respective addressing value to a base address (Column 3 Lines 2-5).

32. Claims 7 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Beard and Bruckert, in view of Ernst et al. (herein Ernst).

33. As per Claim 7, Beard teaches: The method of claim 1,

wherein the determining of duplicates includes:

generating each respective address value for a sequence of addressed locations within a constrained area of memory containing 2^N consecutive addresses using an N-bit value derived from each respective addressing value of the first vector register (Column 7 Lines 30-36),

generating each respective data value of a first sequence of values by combining at least a portion of each respective addressing value of the first vector register to a respective one of a sequence of integer numbers (It is a common practice in computing to "sign extend" values, when their size is not the appropriate size to fit into memory. This entails prefixing the value with 1's or 0's as appropriate. It is likely that data needing to be stored after an operation would need to be sign extended in order to properly fit in the memory, thus requiring the concatenation of some value with a 0 or 1. See Computer Arithmetic Lecture, Pages 11-12),

storing the first sequence of values to the constrained memory area using the generated sequence of respective address values (Column 3, Lines 19-22),

loading a second first sequence of values from the constrained memory area using the generated sequence of respective address values (Column 3, Lines 5-10), and

wherein addresses of the elements from memory are calculated by adding each respective addressing value to a base address (Column 3 Lines 2-5);

wherein the adding includes a floating-point addition operation that produces at least one element of the result vector as an ordered-operation floating point summation of an element of the loaded third vector register and a plurality of respective elements of the original second vector of operand values corresponding to elements of the first vector of addressing values having identical values (Column 4 Line 65 – Column 5 Line 19 shows floating point to be one of the available operations. Also see Column 3, Lines 10-13), but fails to teach:

comparing the first sequence of values to the second sequence of values; and
wherein the loading of the third vector register includes loading elements from locations specified by addressing values corresponding to indications of positive compares from the comparing;

wherein for the storing of the result vector of elements to memory, elements are stored to locations specified by addressing values corresponding to indications of positive compares.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57). One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed, and duplicate data would infer that something unplanned and thus incorrect has occurred. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention.

Ernst teaches selectively replaying instructions that have an invalid bit set (Section 2.4). One of ordinary skill in the art would recognize that selectively replaying instructions means that you would only operate on values that were specified as needing to be replayed, IE, those locations that had a positive compare. Therefore, one of ordinary skill in the art at the time the invention was made would have implemented a selective replay method such as Ernst's in Beards invention to increase performance.

34. As per Claim 15, Beard teaches: The system of claim 14,
wherein the circuitry programmed to determine duplicates includes;
circuitry programmed to generate each respective address value for a sequence of addressed locations within a constrained area of memory containing 2^N consecutive addresses using an N-bit value derived from each respective addressing value of the first vector register (Column 7 Lines 30-36),
circuitry programmed to generate each respective data value of a first sequence of values by combining at least a portion of each respective addressing value of the first vector register to a respective one of a sequence of integer numbers (It is a common practice in computing to "sign extend" values, when their size is not the appropriate size to fit into memory. This entails prefixing the value with 1's or 0's as appropriate. It is likely that data needing to be stored after an operation would need to be sign extended in order to properly fit in the memory, thus requiring the concatenation of some value with a 0 or 1. See Computer Arithmetic Lecture, Pages 11-12),

circuitry programmed to store the first sequence of values to the constrained memory area using the generated sequence of respective address values (Column 3, Lines 19-22),

circuitry programmed to load a second sequence of values from the constrained memory area using the generated sequence of respective address values (Column 3, Lines 5-10), and

wherein addresses of the elements from memory are calculated by adding each respective addressing value to a base address (Column 3 Lines 2-5);

wherein the circuitry programmed to add includes a floating-point adder that produces at least one element of the result vector as an ordered-operation floating point summation of an element of the loaded third vector register and a plurality of respective elements of the original second vector of operand values corresponding to elements of the first vector of addressing values having identical values (Column 4 Line 65 – Column 5 Line 19 shows floating point to be one of the available operations. Also see Column 3, Lines 10-13), but fails to teach:

circuitry programmed to compare the first sequence of values to the second sequence of values; and

wherein the circuitry programmed to load the third vector register loads elements from locations specified by addressing values corresponding to indications of positive compares.

Bruckert teaches comparing two values, one from a primary source, and one from a secondary source, to verify if the data is in agreement (Column 13, Lines 9-57).

Art Unit: 2183

One of ordinary skill in the art would realize the advantage of verifying correct data is being able to recognize if an instruction(s) need to be re-executed, and duplicate data would infer that something unplanned and thus incorrect has occurred. Therefore, one of ordinary skill in the art at the time the invention was made would have been able to attach an error-detecting mechanism such as Bruckert's to Beard's invention.

Ernst teaches selectively replaying instructions that have an invalid bit set (Section 2.4).

One of ordinary skill in the art would recognize that selectively replaying instructions means that you would only operate on values that were specified as needing to be replayed, IE, those locations that had a positive compare. Therefore, one of ordinary skill in the art at the time the invention was made would have implemented a selective replay method such as Ernst's in Beards invention to increase performance.

35. Claims 12 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Beard, Bruckert, and Patterson, in view of Ernst et al. (herein Ernst).

36. As per Claim 12, Beard teaches the method of claim 8, but fails to teach:

further wherein the loading of the third vector register of each processor includes loading elements from locations specified by addressing values corresponding to indications of positive compares from the comparing operation.

Ernst teaches selectively replaying instructions that have an invalid bit set (Section 2.4). One of ordinary skill in the art would recognize that selectively replaying instructions means that you would only operate on values that were specified as

Art Unit: 2183

needing to be replayed, IE, those locations that had a positive compare. Therefore, one of ordinary skill in the art at the time the invention was made would have implemented a selective replay method such as Ernst's in Beards invention to increase performance.

37. As per Claim 25, Beard teaches the system of claim 21, but fails to teach:

further wherein the means for loading of the third vector register of each processor includes means for loading elements from locations specified by addressing values corresponding to indications of positive compares from the comparing operation.

Ernst teaches selectively replaying instructions that have an invalid bit set (Section 2.4). One of ordinary skill in the art would recognize that selectively replaying instructions means that you would only operate on values that were specified as needing to be replayed, IE, those locations that had a positive compare. Therefore, one of ordinary skill in the art at the time the invention was made would have implemented a selective replay method such as Ernst's in Beards invention to increase performance.

Conclusion

38. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure as follows. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

39. Ishizaka (USPN 5,765,009) teaches using barrier synchronization methods in a vector processor.

40. Hashimoto et al. (USPN 6,308,316) teaches using barrier synchronization methods in a vector processor to wait for data.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert E. Fennema whose telephone number is (571) 272-2748. The examiner can normally be reached on Monday-Friday, 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.


Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Robert E Fennema
Examiner
Art Unit 2183

Application/Control Number: 10/643,727
Art Unit: 2183

Page 32

RF



EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100